REMARKS

Claims 1-22 are now pending in the application. Claim 1 is amended to correct a typographical error. Reconsideration as to the patentability of the claimed subject matter is respectfully requested in view of the following discussion.

35 U.S.C. § 103 Rejections

Claims 1-6, 8-11, 13-15, 17-20 and 22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bhagwat et al. (U.S. Patent No. 6,563,517) in view of Halliday (U.S. Patent No. 5,880,740). Applicant respectfully traverses this rejection.

Bhagwat does not teach and every element of the claims as the Examiner has acknowledged in this and previous Office Actions. Halliday does not cure Bhagwat's deficiencies and is inoperable as applied to Bhagwat.

The Examiner has provided numerous cites within Halliday, but none disclose the use of textual references as claimed. The concept of textual references does not exist within Halliday. Halliday is an invention that abstracts the particulars of image creation from the user using, in part, OLE technology. See col. 1, line 58 – col. 2, line 40. For example, Halliday describes creation of a greeting card by a user. See col. 3, line 6 – col. 4, line 59. The greeting card contains "zones" that designate where an image may be placed or changed. Utilizing a mouse, the user selects one of the zones to substitute another image within the zone. The substitute image is retrieved from a library of images suitable for the particular zones selected by the user. When a user has completed selection of images within all the zones (i.e., is done creating the greeting card), a composite image is saved.

The Examiner's first two citations are irrelevant to the instant claims. In these cites (reproduced below), Halliday discloses merely how the individual images that a user selected are stored, e.g., in a database. A Control_ID defines a set of images while the Img_ID points to individual image elements. When an image is created either the composite image may be saved and transmitted to a destination, or the Control and Img IDs may be transmitted. The manner in which Halliday stores its images is irrelevant to the present claims. Although Figs. 8 and 9 show a table of pointers listed together, this is not a representation of how these pointers are actually stored within a database and do not qualify as "textual references" as claimed.

> The Control_ID and Img_ID values seen in FIGS. 8 and 9 preferably take the form of pointers to objects. Objects specified by Control_ID pointers contain instance data specific to each image element (Img_ID object), which in practice would include the zone coordinates XA, YA, XB, YB which are shown separately in FIG. 7 for illustration. Each of the image element objects identified by a Control_ID pointer further contain the identification of the individual image elements; for example, an image element object may contain a pointer to a linked list of Img_ID pointers object pointers. Each object pointed to by an Img_ID pointer may contain a file name for file, or an access key value if the images are stored in a database.
>
> The individual image objects may include a polymorphic rendering method which displays that image on the display in the zone specified by the Control_ID. The rendering method may, if desired, produce an animation within the specified zone, together with an audio accompaniment. An audio the selection of a particular individual image in the greetings zone may cause a different group of images to becomes associated with other zones. This may be readily accomplished by the After function associated with the greeting images which rewrites the Ctable seen in FIG. 8 to substitute different Control_IDs which specify different collections of individual images.
>
> Alternatively the selection of a given image in a given zone may invoke an After function which reorders the images in one or more of the image tables seen if FIG. 9. In this way, the selection of a particular image in one zone may alter the order in which images are presented in another zone. By changing the value J in a given record in the Ctable of FIG. 8, a Before or After procedure may cause the image currently being displayed in another zone to be altered.
>
> When the control values seen in FIG. 8 or 9 are changed to reflect different current images than those actually displayed, the affected zones may be redrawn or the entire composite image may be redrawn by redrawing the individual zones from the bottom up (i.e., K=LastK through K=0 as seen in FIG. 8). Col. 6, lines 25-67.

The Examiner's second citation within Halliday discloses how a word processor or other program may insert an image into a document. Halliday discloses two main methods: inserting the composite image created by the application program by linking the image file to the document file, or making the composite image file a part of the document file. Whether an object is linked or embedded is also irrelevant to the present claims. If an object is linked using well-known OLE technology, the host program merely follows the link to the objects and displays those objects using the program used to create the image to display the image. If a user wishes to edit the image, the host program initiates the program used to create the image to display the image. This process has no relation to the present claims.

> As indicated at 222, such word processors and many other application programs are able to insert a composite image object created by the utility

> program into the container document, either as a linked object (with the composite image data remaining in a separate file) or as an imbedded object (with the composite image data becoming a part of the container document file).
>
> It should be noted, as discussed earlier, that the composite image definition table illustrated in FIGS. 8 and 9 contain image identifiers which may be saved and transmitted instead of the actual image data when the images themselves will are separately stored and accessible using the image identifiers. Thus, the image builder DLL used to select the components of a desired composite image may save those image identifiers to the mass storage unit 204 as a named file. Then, when it is desired to display or print the composite image, the image display (viewer) DLL 202 may be used to reconstruct the composite image previously created. Accordingly, by inserting a composite image OLE object into a container document using the composite image utility program 208 as an OLE server, the user may simply click on the composite image in a container document created by the word processor 220 to invoke the server 208 and the simplified image modification mechanism of FIG. 7 to alter the image. The modified image is then again saved as a modified imbedded or linked OLE object in the usual way. Col. 8, lines 22-47.

With regard to the third Examiner citation, the Examiner incorrectly equates the insertion of an image into an HTML document to the present claims. The complete composite image to be displayed is identified with an HTML tag. A separate program is initiated to follow the link in this tag to retrieve the image (i.e., the SRC parameter). Once the complete image is retrieved, it is displayed. Plugins may be used to enhance the capability of displaying the complete image, or retrieving missing pieces of the complete image.

> As further shown in FIG. 11, the composite images produced in accordance with the invention may also be presented as imbedded images in documents written in Hypertext Markup Language (HTML) and sent from a HTTP web server (as seen at 155 in FIG. 10) or an FTP file server (as seen at 158 in FIG. 10). By way of example, the data which makes up the composite image may be placed in a file having a predetermined MIME data type, the file being designated by a URL which is specified in the SRC parameter of an HTML Imbed tag. An Imbed tag is identified by the browser and passed to a special MIME data handling program 230 which fetches the designated file using the specified URL and displays the composite image at a predetermined position on the page. By way of example, the MIME data handling program may take the form of a Netscape PlugIn, a dynamic code module which is called by the Netscape Navigator web browser to display data of a particular, pre-registered MIME data type when that data is encountered in an Imbed tag or in a file. The structure and function of Netscape Plug-In modules are described in detail in The Netscape Navigator Plug-in Software Development Kit (SDK), Netscape Corp., Netscape Communications Corporation. The Plug-in standard allows third-party developers to extend the capabilities of Navigator by creating new Netscape plug-ins. By incorporating the functionality of the composite image builder and viewer into a

> Netscape Plug-in, the special format used to store composite images using zone definition and image identifiers (FIGS. 8 and 9) may be quickly transferred via the Internet to the web browser which requests the Plug-in display the composite data as a an embedded inline objects.
>
> Alternatively, the composite image may be displayed by the browser using a downloaded Java applet, or the composite image may take the form of an ActiveX (in-process OLE component) capable of translating the composite image data into a displayed image. Note that the individual image elements preferably take the form of files which may exist on the local shared CD-ROM 206, on the local mass storage device 204 (possibly in the browser's cache storage area), or may be downloaded from the web server or an FTP server. An information host source, such as America On Line, an HTTP web site or an FTP server, can include sets of image elements in the initially image set supplied to new subscribers on disk, and download additional images as needed into local storage, thereafter transmitting only the zone coordinates and image identifiers to construct changing composite image combinations from the previously stored individual image elements.
>
> By downloading composite image definition data only, a particular composite image may be constructed in part from previously stored images and, when a given image specified by a component URL is not available, that needed component alone may be fetched from the remote server. The Plug-In, Java or ActiveX image handler may also permit the user to interactively alter the image by executing a routine of the type shown in FIG. 7. Once again, to the extent image elements selected by the user are not currently available in local storage, those elements may be dynamically retrieved from the remote server. Col. 8, line 62 – col. 9, line 52.

Halliday is an invention for creation of an image by a utlity which abstracts the particulars of image creation from a user. Once created, the pointers to particular images are stored in a database. Halliday does not disclose any capability to analyze, for example, markup language encoded in a web page document to search for adjoining textual references as claimed.

Halliday is simply not applicable to the present invention as claimed or Bhagwat. As stated in Applicant's previous responses, the Examiner is incorrectly interpreting the claim elements as evident by the language used in the instant Office Action. In suggesting that Halliday teaches the necessary elements, the Examiner states:

> It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of *Bhagwat et al* with *Halliday et al* for the purpose of transcoding and linking directly adjoining images from a webpage so as to generate a transcoded composite image with adjoining references; because this allows for the transcoding of an entire image by grouping the image segment in order to scale the composite image according to the display properties of limited-display device in order to properly render it on a limited-display device. Office Action, page 4 (emphasis added).

However, the claims do not state "linking directly adjoining images from a webpage so as to generate a transcoded composite image with adjoining references." This interpretation is not what is disclosed and claimed. The present claims recite searching a web page document for sequences of textual references directly adjoining each other not searching for directly adjoining images. Whether the images adjoin each other is irrelevant. For example, in HTML coding, an image or formatting object is described by the use of various tags, for example, <img> or <table>. For example, Figs. 9 & 10 of the instant specification, reproduced below, reveal such coding.

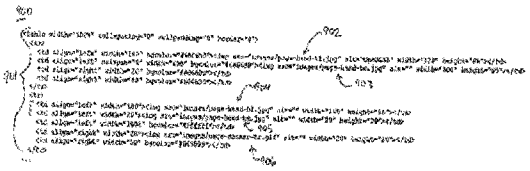

figure 9

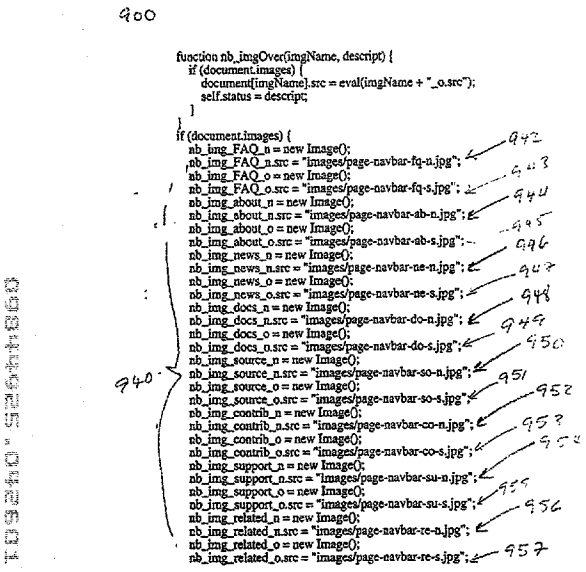

FIG. 10

As seen in Fig. 10, the images described are navigation images, which in some cases visually do not directly adjoin each other when displayed as a typical web page. However, the textual references to these navigation images are directly adjoining each other in the document describing the web page. The present invention as claimed searches for these sequences of textual references and processes them according to the claims. What the images visually look like is unknown and irrelevant to the process because the HTML coding already defines location and content. What is relevant is that the HTML tags (i.e., the textual references) are in a sequence directly adjoining each other) and processed as claimed. In some instances, the images referred to may not even be displayed at the same time (e.g., in the case of a "mouseover" in

which one image is replaced or supplemented by another when the user places the mouse pointer over the original image). In this example, the images do not directly adjoin each (they replace each other), but the textual references in the web page document describing the web page do adjoin each other. Thus, the teachings of Halliday and the instant specification are quite disparate.

Moreover, and as stated in the previous Office Action, the end result of the claims is not "a transcoded composite image with adjoining references," or "an entire image by grouping the image segments in order to scale the composite image," but "a composite image" to be scaled and displayed as claimed so that a particular device does not need to waste precious resources transcoding and transmitting each image individually.

As such, Halliday (like its predecessors, Horie and Katayama, cited in previous Office Actions) discloses none of the above elements. Combining Halliday with Bhagwat makes either Halliday or Bhagwat inoperable, or at best, adds nothing to Bhagwat. Accordingly, the Applicant respectfully requests this rejection be withdrawn.

Claims 7, 12, 16, and 21 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Bhagwat in view of Halliday and further in view of Robotham et al (U.S. Patent No. 6,704,024). Applicant respectfully traverses this rejection. As discussed above, Halliday does not cure the deficiencies of Bhagwat. As such, the combination of Robotham does not cure the remaining deficiencies. Accordingly, the Applicant respectfully requests this rejection be withdrawn.

Conclusion

Applicant submits the claims are in condition for allowance and respectfully request that the Examiner reconsider the outstanding rejections. The Examiner is invited to telephone the undersigned representative if an interview might expedite allowance of this application.

Respectfully submitted,

BERRY & ASSOCIATES P.C.

Dated: October 1, 2009

By: /Shawn Diedtrich/
Shawn Diedtrich
Registration No. 58,176
Phone: 480.704.4615

Correspondence Address
Cust. No. 49637
Berry & Associates, P.C.
9255 Sunset Boulevard, Suite 810
Los Angeles, CA 90069
Phone: (310) 247-2860
Fax: (310) 247-2864